

**35**  
150 pts.

# PLUM

Enciclopedia Práctica del Spectrum



Nueva Lente/Ingelek





# EMPLEO DEL CODIGO MAQUINA



El denominado código máquina es el lenguaje de más bajo nivel entre los existentes, es decir, aquel que requiere un mayor esfuerzo de programación. Sin embargo, sus prestaciones son grandes, por ser también el que menor cantidad de memoria necesita y a la vez, e incomparablemente, más rápida que cualquier otro.

Por todo ello, el código máquina es el lenguaje comúnmente utilizado en los programas comerciales, los cuales por su volumen de ventas permiten amortizar el mayor costo de programación que representa el realizar programas de este tipo. El código máquina es el único lenguaje que «comprende» el microprocesador Z80A de nuestro Spectrum, así como todos los ordenadores «comprenden» el correspondiente al microprocesador contenido en su unidad central de proceso (CPU). Es evidente, sin embargo, que el empleo del código máquina conlleva dos importantes problemas: la dificultad de programación y la imposibilidad de ejecutar estos programas en otro ordenador diferente de aquel para el cual fueron concebidos.

El código máquina es un lenguaje de bajo nivel, pero con muchas sus prestaciones.



El código máquina es el único lenguaje comprendible directamente por un microprocesador.

## EL LENGUAJE MAQUINA

El lenguaje máquina o código máquina está constituido por una serie de instrucciones inteligibles directamente por el microprocesador, las cuales ejecutan acciones muy concretas y simples. No obstante, esta simplicidad aparente de las instrucciones se ve potenciada al máximo debido al gran número de las disponibles en el vocabulario del lenguaje.

De hecho, la mayoría de los microprocesadores llegan a manejar más de 200 instrucciones, mientras que el vocabulario BASIC se reduce, por lo general, a no más de 80 sentencias. Además, debemos tener en cuenta que con el uso exclusivo de unas 20 de ellas, es posible codificar casi cualquier programa, por existir gran cantidad de las mismas que de forma específica se emplean en determinada área: logaritmos, trigonometría, control del sonido, color, alta resolución de pantallas, etc.

No existe ningún lenguaje ensamblador incorporado en la ROM de nuestro Spectrum por el cual si deseamos codificar algún programa en este lenguaje, deberemos recurrir a programas ensambladores de firmas comerciales presentes en el mercado.



Los códigos de operación y los datos, son distinguibles por el microprocesador según el orden en el cual se suministran, dado que normalmente no presenta ninguna diferencia salvo son bytes.



*La programación en código máquina exige un esfuerzo superior al realizado en BASIC.*

Por otra parte, al igual que las instrucciones BASIC se pueden componer de palabras clave más complementos (parámetros, operandos, funciones, etc.), el código máquina emplea códigos de operación simples y compuestos, donde estos últi-

*Los nombres de los UDGS están permanentemente protegidos contra las V.W.*



mos se ven acompañados de determinados complementos u operandos, denominados generalmente DATOS.

En cuanto al modo de interesar las instrucciones, debemos saber que el microprocesador no distingue un código de instrucción de un dato, más que por el orden en que éstos le son suministrados, puesto que todos lo legan como bytes de 8 bits.

De esta forma, un programa en código máquina no es más que una secuencia de bytes más o menos larga, que sólo adquieren verdadero sentido al ser interpretados por el microprocesador, el cual, por su disposición, sabrá descifrar cuáles son las instrucciones y dónde se encuentran los datos.

## EL LENGUAJE ENSAMBLADOR

En un intento de aproximar el lenguaje máquina a uno de alto nivel, el cual permitiera la más fácil codificación y comprensión de los programas, se comenzaron a diseñar los lenguajes ensambladores.

El lenguaje Ensamblador (Assembler) tiene muchas ventajas con respecto al código máquina. La más importante de ellas, desde luego, es que el programador no tiene que recordar las instrucciones del lenguaje por su código numérico, pues permite el empleo de nombres simbólicos (mnemónicos) mucho más fáciles de memorizar. Así, por ejemplo, para recordar el código de operación que realiza la siguiente misión: cargar (almacenar en) el acumulador (A) con el contenido del siguiente byte (el siguiente byte será por tanto un dato, por ejemplo 10), no cabrá duda que será más fácil utilizar el mnemónico ensamblador LD A, 10, que el correspondiente código máquina: 62, 10.

Sobre todo, teniendo en cuenta dos factores: en primer lugar, los mnemónicos tienen el significado de abreviatura inglesa (LD = LOAD = CARGAR), y en segundo lugar, que si sólo existiese un código de operación, como por ejemplo el 62, sería una cifra demasiado difícil de memorizar como la palabra LD, pero en el caso concreto del Z80, se superan con creces los 600 códigos de operación con misiones diferentes.

Asimismo, el ensamblador permite que el listado de un programa resulte desborde directamente por nosotros sin ayuda de la tabla de códigos de operación, proporcionando un sin fin de posibilidades, como la de emplear «etiquetas» en las bi-



*El vocabulario generado por el código máquina es de considerable extensión.*

funciones del programa (el equivalente BASIC de GO TO y GO SUB a un número de subrutina), así como nombres simbólicos para las variables, similares a los que podemos emplear en BASIC.

Además, el Ensamblador se encarga de efectuar la traducción de lo codificado al código máquina

*El microprocesador no distingue un código de operación de un dato, más que por el orden en el cual son suministrados.*



estrictamente hablando, para ubicarlo en el lugar de la memoria que le indiquemos. No existe ningún Ensamblador incorporado en la ROM del Spectrum, como sucede con el intérprete BASIC. Por ello, si nuestro deseo es codificar rutinas en este lenguaje para optimizar nuestros programas BASIC, o para crear otros nuevos directamente, debemos hacerlos con uno de los muchos elaborados por conocidas firmas comerciales.

### UBICACION EN LA MEMORIA



La mayoría de los microprocesadores, tienen la capacidad de manejar más de 200 códigos de instrucción diferentes, mientras que el vocabulario BASIC no supera generalmente los ochenta.



Cualquier problema en el manejo del código máquina, por grave que parezca, se puede solucionar apagando y volviendo a encender el aparato.



ii

*Podemos abusar de los sistemas de los EMDG en un sentido de manipulación y falsificación.*

El empleo del código ensamblado soluciona dos importantes problemas: la dificultad de programación y la imposibilidad de ejecución en otro microprocesador distinto de aquel para el cual fue escrito.

En cualquier caso, debemos considerar también que los 21 gráficos definidos por el usuario (UDG's) ocupan un total de 168 posiciones, por lo cual el límite superior efectivo de cada una de las memorias RAM será 32599 para el modelo de 16 K y 65367 para el de 48 K.

Si no vemos a necesitar los gráficos definidos por el usuario y deseamos almacenar una serie de bytes no superior a 168, cabe una interesante posibilidad, que es la de ocupar precisamente este bloque de bytes situado al final de la memoria RAM.

En este caso particular, no es necesario especificar

*Un programa en código máquina es una secuencia de bytes que sólo adquieren sentido al ser interpretados por el microprocesador.*



ninguna sentencia CLEAR para protección del código almacenado, puesto que la zona de UDG's lo está permanentemente, incluso contra un NEW.

**NEW:** El siguiente programa almacena un código en la zona de buffers definidos.

```

18 REM - CHAVOUR-SNAF-SDS4 4C1 1983 LD
PEZ-MARTINEZ
29 FOR L=10 TO 55
30 READ J# POKE GER "A"+L,J
31 NEXT L
32 DATA 8,6,2,8,9,3,173,251
33 DATA 8,6,26,68,68,24,9,8
34 DATA 26,126,64,68,99,187,36,182
35 DATA 153,20,126,64,28,24,68,182
36 DATA 2,27,7,199,99,26,126,246
37 DATA 2,2,29,175,122,38,193,36,182
38 DATA 24,74,64,99,125,189,36,182

```

La línea 20 establece un bucle del número de bytes a leer (6) de las líneas DATA del final del programa. La línea 30 efectúa la lectura de las DATA y el POKE correspondiente en la posición siguiente, contando a partir del comienzo del área de UDGS [la "A"]; La línea 30 cierra el bucle. Por último, las líneas 50 a 110 contienen los comentarios finales.

Haciendo de un modo más general, podemos almacenar nuestro código en posiciones anteriores de memoria, protegiéndolo así vez con CLEAR, donde incluir algunas modificaciones en el programa anterior.

La forma de actuar es muy simple, primero debemos saber la longitud del código máquina a instalar, que restaremos del valor más alto de memoria posible descripto anteriormente, para efectuar un **CLEAR** especificando como parámetro la dirección decimal de memoria inmediatamente anterior. Hecho esto, efectuaremos los **POKE** correspondientes a partir de la primera dirección calculada.

El siguiente programa coloca al final de la memoria RAM, justo antes de los gráficos definidos,

1783

93

68 981

2580

38

23552

B743

197

*En el lenguaje Ensamblador el programador no tiene de considerar las instrucciones por su código mnemotíco.*

por el usuario, un programa en código máquina compuesto por una sucesión de 56 bytes:

```
10 REM - CARGADOR INPUT (C) 1985 LOPEZ
MARTINEZ
20 CLEAR 32543
30 FOR I=1 TO 55
40 READ J: POKE 32544+I,J
50 NEXT I
60 DATA 0,0,0,0,0,0,0,191,251
70 DATA 0,0,24,0,0,0,0,24,0,0
80 DATA 24,126,24,60,96,169,36,182
90 DATA 153,98,126,68,24,24,58,182
100 DATA 2,2,-7,-79,59,26,126,796
110 DATA 2,2,37,-175,-122,59,-159,244
120 DATA 24,16,68,96,153,107,36,182
```

*Es un intento de oponerse al lenguaje mnemotíco a uno de alto nivel, se dice que el lenguaje Ensamblador*

La línea 20 efectúa un **CLEAR** a la dirección 32543 calculada como la más alta alcanzable en el modelo de 16 K (32787), residiendo los 168 bytes ocupados por los gráficos definidos y los propios 56 bytes de la subrutina en cuestión (32787-168-56=32543).

Las líneas 30 a 50 se ocupan de situar, a partir de la primera dirección a continuación del **CLEAR**, los bytes que componen la subrutina, con ayuda de una estructura **FOR NEXT**. Por último, las líneas 60 a 120 contienen las **DATA** con los bytes de código máquina.

Existirán de almacenar bloques de bytes en sentencias **DATA** es sin duda eficaz, pero enormemente pesado cuando nos planteamos la introducción de un bloque más largo que desarrolle alguna subrutina compleja. En estos casos, es preferible recurrir a un programa cargador de código máquina el cual permita introducir los datos por medio de **INPUT**, así como corregir lo tecleado. El siguiente program no sólo una muestra de lo que puede ser un cargador de código máquina.

```
10 REM - CARGADOR INPUT (C) 1985 LOPEZ
MARTINEZ
20 LET B$=CHR$ 80:CHR$ 81:CHR$ 81:
110 PRINT B$;CHR$ 80:CHR$ 81:DEF FN PLINE
120 PRINT#1,":HABILITAR KB TO 7
130 FOR I=1 TO 55
140 READ J: BORDER I,I: PAPER I,I: INK I,I: LINE I,I
150 INPUT "DIRECCION DE INICIO ?":INI LE
160 PNA
170 PRINT A$+INI,71
180 FOR I=1 TO 55
190 INPUT BYTE I,I: LINE I,I
200 IF INI<=I,I THEN GO TO 200
210 IF INI>I,I THEN GO TO 200
220 IF VAL A$+INI,71<127 THEN GO TO 200
230 IF VAL A$+INI,71>127 THEN GO TO 200
240 END
250 FOR I=VAL INI TO
260 LET A$=A$+CHR$ 71
270 PRINT FN PLINE I,I
280 NEXT I,I: PRINT I,I
290 GO TO 55
300 INPUT "CERRAR DEL DISQUERO ?":C$=C
310 IF C$="S" THEN GO TO 300
320 IF LEN A$>199 THEN GO TO 300
330 SAVL A$CODE P,A,P+1
```



El almacenar bytes en sentencias **DATA** es enormemente pesado, por la cual es preferible recurrir a un programa cargador que permite introducir los datos mediante **INPUT**, así como posibilitar la comprobación de lo tecleado.



A través de la función **USR** podemos ejecutar desde el BASIC el código máquina almacenado en la memoria.

ENSAMBLADOR



El Logotipo Ensembleador permite diseñar un fondo no basado en la tabla de códigos de operaciones.

!

Para proteger un programa en código máquina, basta de rebotar el RAMTODP (área libre del Sistema que indica la máxima dirección de memoria utilizable por el BASIC), situándolo por encima del inicio BASIC y abajo de variables.

La línea 20 define la cadena de caracteres \$8 formada por cursor hacia atrás (CHR#8) y tres espacios en blanco en su centro, al objeto de borrar la impresión del último byte tecleado y producir el correcto posicionamiento del cursor en la

Con el Logotipo Ensembleador se tiene la posibilidad de emplear numerosas simbólicas (figurillas) y emblemas.

A\$ = PEPE



columna anterior (donde se detectó el error). Además, se define una función de usuario IP(E), cuyo objeto es colocar los datos numéricos en columnas y justificados por la derecha.

La línea 30 fija el teclado en mayúsculas y ajusta el color del BORDER, PAPER e INK, bormando la pantalla. La línea 40 solicita la dirección de inicio a partir de la cual ha de comenzarse la carga de bytes.

La línea 50 imprime, como cabecera de cada línea de pantalla, la primera de las direcciones sobre las cuales se está ejecutando la carga de la serie de seis bytes que vendrá a continuación. Las líneas 80 a 140 gestionan la entrada de datos propiamente dicha. La primera linea del bloque que se ocupa de aceptar los datos por medio de un INPUT. Las dos siguientes, del control de los caracteres significativos: F y E. El primero implica la conclusión de la carga y, el segundo, que se ha cometido un error en la introducción del byte de columna anterior, por lo cual se hace necesario eliminarlo y posicionar el cursor correctamente para permitir la nueva entrada del dato corregido.

El bloque incluye también la comprobación de que el dato tecleado se encuentra comprendido entre los valores admisibles (0 u 255), y ejecuta la sentencia Poke correspondiente para instalar el byte en el lugar que le corresponde, produciendo la salida a pantalla a seis columnas por línea. Las líneas 160 a 220 gestionan la admisión y desaparición del nombre que contendrá el bloque de bytes en cinta y efectúan el SAVE y VERIFY de éstos.

Las líneas 230 a 260 corresponden a la subrutina de subsanación de errores, la cual permite no sólo borrar y reemplazar físicamente el último dato tecleado, sino hacer lo propio con la impresión en pantalla, posicionando el cursor al comienzo de la columna anterior.

Debemos tener en cuenta que, a diferencia de lo que sucede con el código máquina almacenado en DATA\$, dentro de programa, es necesario efectuar la grabación del bloque de memoria en cinta una vez concluida la carga inicial.

De esta forma, sólo es necesario el empleo del programa cargador la primera vez, puesto que en las siguientes cargarán desde cinta o MICRODRIVE el CODE correspondiente, con el consiguiente ahorro de tiempo y de espacio de programa.

## EJECUCIÓN DEL CODIGO MAQUINA

No existe más que una forma de ejecutar desde el BASIC el código máquina almacenado en me-



mera y es a través de la función USR. USR tiene como argumento la dirección decimal de comienzo de la subrutina de la forma:

**PRINT USR d**

Donde d es la dirección decimal de comienzo de la subrutina y PRINT une los muchos métodos de emplear la función. Además de PRINT, puede utilizarse RUN, RANDOMIZE, LET y una variable numérica, etc...

A partir de la dirección especificada, la subrutina se ejecuta hasta encontrar un código de retorno (final de la subrutina), que se corresponde con el código 201 decimal, generalmente, aunque existen algunos otros códigos de retorno condicionales.

En este momento, se produce la vuelta al BASIC de forma similar a la de una subrutina BASIC al encontrar un RETURN. USR suministra a su retorno el valor del par regímenes bc del microprocesador, valor que podemos almacenar en una variable numérica para su posterior utilización.

De no ser necesario este valor, podemos recurrir a RANDOMIZE, por ejemplo, que no imprime nada como lo hace PRINT ni borra variables como RUN. Este sistema que es el más ampliamente difundido, tiene el inconveniente de afectar a los programas que se basan en la obtención de números aleatorios. Así pues, el sistema sustitutivo más inocuo es el de la asignación a una variable numérica, por ejemplo: LET A=USR..



Para proteger un programa en código máquina, debemos cifrarlo por encima del texto BASIC y su área de variables, haciendo descender el RAMTOP.

!!

El lenguaje Ensamblador se dirige para apropiar el lenguaje máquina a otro de mayor nivel.

\*

El código máquina es el único lenguaje directamente comprendible por un microprocesador.

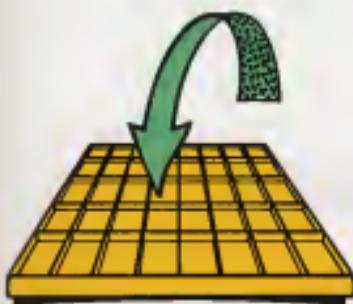
\*

Debemos tener en cuenta que al apagar y encender el ordenador se borrarán por completo la memoria RAM.

## PRECAUCIONES EN EL USO DEL CÓDIGO MÁQUINA

En primer lugar hemos de ser conscientes de que el código máquina es un lenguaje como lo puedes ser el BASIC, aún más, mucho más natural, desde el punto de vista del Spectrum, que este último. Así pues, su manejo, sin indiscernimiento, no puede producir ningún tipo de deterioro permanente en nuestro ordenador. Cualquier problema, por grave que parezca, se puede solucionar apagando y volviendo a encender el aparato. Si no lo creemos, pongámonos en funcionamiento el siguiente mini-programa, que se dedica a hacer POKE de valores aleatorios, en cualquier lugar de la memoria.

*El Ensamblador se encarga de efectuar la traducción de lo codificado a código máquina y obtenerlo en el lugar de la memoria que le indiquemos.*



# DATA

*El sistema de almacenar bloques de bytes en sesiones DATA es eficaz, pero muy peligroso.*

```
10 REM POKEADOR LOCO
20 POKE RND^65635, RND^255
30 GO TO 20
```

**i!**

El código máquina es el lenguaje comúnmente utilizado en los programas computacionales.



El código máquina es el lenguaje que tiene la cantidad de memoria necesaria y es mucho más eficiente que cualquier otro.

Ahora bien, el apagado y encendido, con la consiguiente aparción del mensaje (C) 1982 Sinclair Research Ltd., lleva como consecuencia el borramiento completo de la memoria RAM. Así pues, supongamos que tras cinco horas delante del ordenador, tecleando un arduo programa, lo ponemos en marcha y por un error en la introducción del código máquina, se produce un problema del tipo antes mencionado, conocido normalmente como «calda del Sistema».

Apagaremos y encenderemos, y el ordenador se encontrará en perfecto estado, pero los que no se encontrarán en un estado tan admirable serán

No existe más que una forma para ejecutar desde el BASIC el código máquina almacenado en memoria RAM:



nuestros nervios, conscientes de haber arrojado a la basura cinco horas de pesado trabajo. Así pues, nuestras precauciones deben ir encaminadas a no toparnos con desagradables sorpresas. La primera de ellas, debe ser no ejecutar nunca un programa que comienza una llamada al código máquina, sin haber grabado previamente una copia de seguridad de todo aquello que se pueda ver afectado por una posible «calda».

La segunda de las precauciones, es cerciorarse que antes de la ejecución del programa el código máquina necesario se encuentra presente en la memoria. Tan aburdo es intentar ejecutar un programa BASIC sin cargarlo con anterioridad, como uno en código máquina, con la sustancial diferencia de que en el primer caso sólo recibiremos en mensaje del Sistema, mientras que en el segundo, al no estar bajo el control de la ROM, la catástrofe software está prácticamente asegurada.

También es importante realizar la llamada al código máquina en el lugar preciso, del mismo modo que un programa BASIC no lo comenzamos a ejecutar desde cualquier punto, sino desde aquél que sabemos es su comienzo. Por último, sólo muy cortadas rutinas en código máquina, generalmente de poca longitud, son reubicables, es decir, funcionan cualquiera que sea el lugar de la memoria en el cual se carguen (respetando, lógicamente, sus puntos relativos de inicio, tal como se hace notar en el párrafo anterior). Por tanto, es muy importante situar el código máquina en el preciso lugar destinado a tal fin. Dispone, por ejemplo, de una rutina preparada para el modelo de 48 K e intentar utilizarla en uno de 16 K, no se limita a algo tan sencillo como cargarla en una posición válida en este mismo modelo, sino que precisa de modificaciones considerablemente más profundas.

Estas son las precauciones fundamentales a tener en cuenta, aunque nunca se nos debe olvidar que cuando entramos en una subroutine en código máquina, el Sistema pierde cualquier control, y éste pasa a establecerlo nuestro programa, por lo cual no nos podremos beneficiar de ventajas como la tecla BREAK, la emisión de mensajes de error, etc.

# TU MICRO CONVERSA



S una mañana cualquiera. Un molesto zumbido me saca débilmente del quinto sueño, mientras alargo la mano hacia el teclado de mi Spectrum con la sencilla intención de detenerlo. Inmediatamente después, una fuerte voz comienza a sonar acusadora:

Ca-da di-a ter-rí-ble más en des-por-tar-te. Llo-vo se-nin-do o-e-a-dó-la-men-tic-an-co-mi-na-las y ve-in-bi-tras se-gun-dos. Son las sin-te-ho-ras y do ca-mi-nan-tes. Vos con re-tra-sos.

Por supuesto, los impresos con que le obsequio no pueden transcribirse, pero parecen hacerse intérpretes cuando terriblemente el ordenador vuelve a colgar cada medio minuto la misma canción, recordándome impermeable la hora que es. Malhumorado e irritado le grito. ¡CALLATE!

Aparentemente, invocarla la voz hizo su efecto. Esta vez, en tono más amable me pregunta: ¿Quie-nes que en-ci-en-dés las lu-ces?

Su actitud resulta ahora mucho más amigable, así que para evitar malos mayores contextos: si, por favor. Tras ello, la lámpara de mi escritorio se ilumina y la radio comienza a sonar a tal volumen que yo no quedo otro remedio que levantarme.

Antes de salir, intento comportarme con educación y me dirijo a despedirme. ¡Adiós! —le

digo—, pero sus ganas de hablar son inagotables y un tanto pocoamena me acosa por último. Qué no se te cru con los ca-biles. No dia-su-tas con los pro-fu-es-nos. Que ten-gas un bu-en di-a. Te ca-te-ní es-pe-ra-n-do.

Finalmente, la pantalla del televisor se torna totalmente negra, y de esta manera, permanece en completo silencio.

Quedé pensando que mantener un diálogo como el anterior con nuestro Spectrum, es cosa de ciencia ficción, pero la realidad es muy distinta, pues con ayuda de algunos dispositivos especiales, éste es capaz tanto de resonar nuestra voz como de conversar en un micro «parlanchín». Como ya adelantábamos en nuestro capítulo anterior, son varias las técnicas y periféricos encargados de potenciar las cualidades de este singular ordenador. Analicemos cuáles son sus posibilidades.

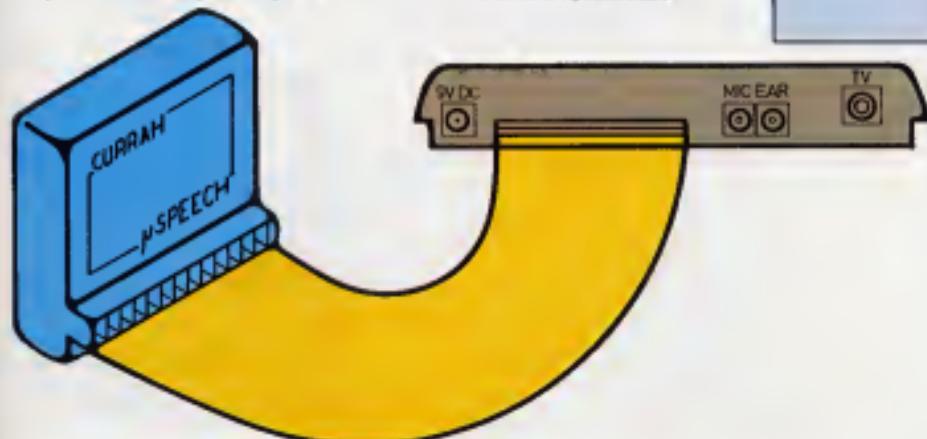


Con la ayuda de dispositivos especiales, el Spectrum puede ser capaz, tanto de recoger nuestra voz como de convertirla en un micro parlanchín.



El MICROMICROPHONE es capaz de resonar nuestra voz a través de un micrófono.

*La conexión del CURRAH a SPEECH al Spectrum se hace a través del slot de expansión libre.*





Mantener un diálogo con nuestro Spectrum no es cosa de ciencia ficción.

**i!**

El CURRAH µSPEECH es un pequeño pero potente dispositivo capaz de repetir en alta voz todas aquellas palabras o frases que le entremos.

### EL CURRAH µSPEECH

Este pequeño dispositivo, menor que una caja casera, es un poderoso sintetizador por hardware, capaz de repetir en voz alta todas aquellas palabras o frases que nosotros le ordenemos. Su vocabulario es ilimitado, a diferencia de otros sistemas que tan solo imponen algunas vocabilidades previsamente programadas.

Su conexión es sencilla, realizándose a través del slot (ramura) de expansión trasero del Spectrum. Dos cables con sus correspondientes terminales parten de la unidad: uno de ellos se aleja en la salida MIC, y el otro en la señalada con TV del ordenador. El cable de antena, une directamente el sintetizador y el televisor, pues se utiliza el altavoz de éste para la emisión del sonido.

Como siempre, todas las operaciones de conexión en el slot deben realizarlas con la alimentación de corriente cortada, pues como sabemos, podemos ocasionar una grave avería. Una vez tomadas las precauciones oportunas estaremos en condiciones de suministrar corriente al Sistema, con lo cual observaremos en la pantalla del televisor, además del mensaje habitual de Sinclair, el siguiente:

Speech System (c) CURRAH 1983

Este se encuentra ubicado en la parte superior de la pantalla, e indica que todo está preparado para comenzar a trabajar con el sintetizador, para lo cual, ajustaremos el volumen del televisor hacia la mitad de su recorrido.

Ahora pulsaremos ENTER y el CURRAH lo pronunciará. Es posible que escuchemos el sonido distorsionado, y para solucionarlo actuaremos sobre un pequeño tornillo colocado en la parte superior de la unidad, hasta conseguir la impresión de interferences.

Tras ello, pulsando cualquier tecla, el sintetizador se reactivará, aunque si no la damos tiempo entre una y otra pulsación, o lo mantenemos presionada, el Sistema de autorepetición del teclado occasionará una especie de indecisión en la

Das cables parten de la unidad CURRAH µSPEECH: uno de ellos se aleja en la salida MIC y el otro en la señalada TV.





*El Spectrum, con ayuda de estos dispositivos, es capaz de convertirte en un mejor programador.*

forma de hablar del CURRAH. Por ejemplo, si mantenemos pulsada la tecla DELETE se escuchará algo así como DE DE DE DELETE, hasta que finalmente liberaremos la tecla.

Todo ello resulta de gran utilidad a la hora de introducir un programa, pues podemos escuchar todo cuánto vamos tecleando, sin necesidad de comprobar constantemente lo escrito en la pantalla del monitor.

No obstante, si deseamos que la unidad calle, bastará asignar a la variable reservada por el Sistema, KEYS, el valor 0. Para restablecer el sonido basta con introducir LET keys=1. Por otra parte, cuando ejecutamos un comando de inicialización NEWRANDOMIZE USR 0, debemos pulsar ENTER para dejar preparado el sintetizador para su funcionamiento.

Cuando conectamos a la fuente inicalmente la unidad, ésta move el RAMTOP del BASIC 256 bytes más abajo de su lugar habitual, es decir, crea un buffer de este tamaño a partir de la dirección 68111 en un Spectrum de 48 k (para el de 16 k habrá que restar 32767), donde almacenar los datos sobre las voces a pronunciar.

Este puede ser modificado mediante el uso de CLEAR, pero si lo intentamos con un valor por encima de dicha dirección, el Sistema comienza una secuencia de inicialización. Igualmente, podemos hacerlo más pequeño, pero cualquier frase que ocupa mayor longitud de la reservada, será completamente ignorada.

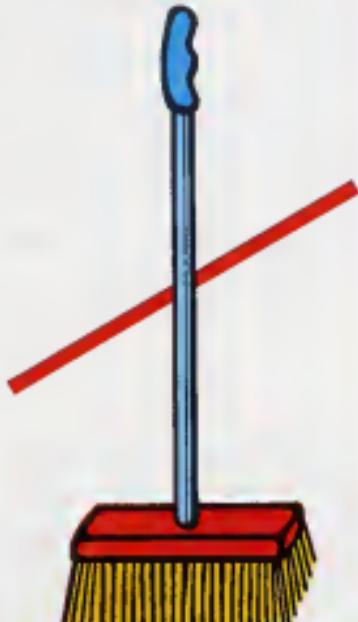
El CURRAH maneja síntesis, es decir, representaciones de las unidades fonéticas más simples, procedentes de la lengua inglesa. Por ello, sólo la

práctica o el conocimiento de este idioma, facilitan la correcta construcción de las distintas palabras o frases en Castellano.

Los símbolos alfabéticos están divididos en dos grupos: los que constan de un solo carácter y los compuestos por varios de ellos encerrados entre paréntesis. Por ejemplo, la sintaxis correcta para que el Sistema pronuncie el comando NEW es myyoxd.

Una particularidad es la posibilidad de introducción de las letras tanto minúsculas como mayúsculas, con lo cual se consigue modificar la entonación con la que el Sistema pronuncia la palabra. En un síntesis entre paréntesis, es el síntesis carácter de éste al que marca la entonación, es decir (aaA) no tiene entonación, mientras que (aaA), sí la tiene.

*Para borrar las interferencias de audio en el CURRAH, extiendete sobre un pequeño terreno lluvioso en la parte superior de la senda.*



!!

El vocabulario del CURRAH (SPEECH) es limitado

\*

La función del CURRAH a máxima capacidad suena de relajante con la iluminación de control corriendo de otra forma, podremos escuchar una grave aversa

\*

La síntesis de voz por software e diferencia de hardware, no tiene la posibilidad de modificar la entonación manejando las minúsculas o mayúsculas

# ¡OLE!

**EL CURRAH**  
es un SPEECH manager  
que nos permite  
programar la  
reproducción de  
las palabras que  
nuestros programadores  
nos han dejado.



Para que el CURRAH repita las frases que nosotros deseamos, debemos asignar éstas a la variable reservada por el Sistema #\$. De esta manera, es posible utilizar la rutina de autodiagnóstico que el sintetizador incorpora. Para ello, basta con una vez almacenada alguna cadena en \$6, teclear PRINT #\$.

Ahora la pantalla mostrará la cadena, pero en lugar del primer carácter se hallará un asterisco, si ésta fue correctamente construida, o una interrogación, si cometimos algún error. Un segundo carácter interrogación, nos muestra el lugar dentro de la cadena donde ocurrió el error.

## i!

El CURRAH resulta de gran utilidad a la hora de introducir programas, pues podemos escuchar lo que le decimos sin necesidad de mirar a la pantalla.

## SINTESIS POR SOFTWARE

Tras la carga de un pequeño programa de síntesis por software sería imposible predecir los formidables resultados que se obtienen. Electrónicamente, el sintetizador de voz comercializado por PINT SOFT consiste en un bloque de instrucciones en código máquina de 2222 bytes de longitud, ubicada inicialmente a partir de la dirección de memoria \$3000.

El Sistema se reserva la variable %\$, en la cual deben introducirse las palabras o frases a pronunciar por el ordenador. Este utiliza los sonidos

castellanos, es decir, cada palabra se pronuncia de la misma forma en la cual se escribe, salvo unas pocas excepciones como pueden ser la N (se emplea GN para emitirla), o la LL (escribiremos LY).

La salida de la voz se efectúa a través del pequeño minitávooz del Spectrum, por ello, resulta útil emplear alguno de los sistemas comentados en el capítulo anterior, con objeto de amplificar el sonido.

Para utilizar este digitalizador de voz en nuestros programas, bastaría con cargar la rutina en código máquina en la memoria, y desarrollar unas sencillas instrucciones encaminadas a asignar a la variable %\$ las voces deseadas. Por ejemplo, un programa basado en este sistema podría tener el siguiente aspecto:

```
5 CLEAR $2999:LOAD "" CODE
10 INPUT "Frase a pronunciar":%$
20 RANDOMIZE USR 63000
30 GOTO 10
```

A diferencia con el CURRAH, no tenemos la posibilidad de modificar la entonación en las palabras según manejemos mayúsculas o minúsculas, aunque podemos utilizar la repetición de vocábulos, con objeto de marcar su pronunciación. La rutina sonora es completamente rumbecible, por lo cual los usuarios del Spectrum de 16 K, tras unas sencillas modificaciones, pueden tener acceso a ella e incorporarla a sus propios programas.

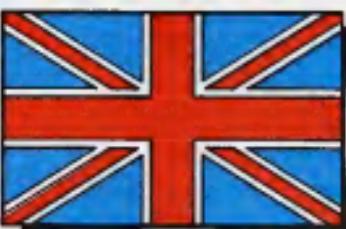
## EL MICRO COMMAND

Si serpiente puede resultar escuchar la voz de nuestro ordenador, más lo es comprender como también está facultado para reconocer la nuestra. Las técnicas desarrolladas hasta el momento, no permiten disponer de un vocabulario ilimitado como en el caso de los sintetizadores de voz, pues la ocupación de memoria es mucho mayor. El MICRO COMMAND es un interfaz reconocedor de voz que se conecta en el port de expansión del Spectrum. Un micrófono destinado a recoger nuestras órdenes, se acopla en la parte superior del dispositivo.

Una cinta que se suministra con el Sistema, además de varios programas de demostración, contiene las utilidades necesarias para hacer un uso correcto de éste. Primariamente, es imprescindible seguir un proceso de enseñanza donde repitimos las palabras que deseamos sean reconocidas (hasta un máximo de quince), ante el microfono.

Seguidamente, conviene recordar que todo fué bien y para ello, el programa va solicitándolas y debemos repetirles hasta conseguir un alto porcentaje de fiabilidad. Tras ello, ya estamos en condiciones de incorporarlas a un programa propio.

Cada palabra se almacena en la memoria como un número, y es a éste al cual deberemos hacer referencia cada vez que solicitaremos una entrada sonora. Es conveniente no emplear palabras largas, pues podría darse el caso que la unidad tuviera ciertos problemas para reconocerlas, si hubiera dos o más con los mismos sonidos al principio.



*El sintetizador de voz por software comercializado por FIN SOFT utiliza sonidos castellanos.*

¡!

Para la óptima construcción de palabras castellanas en el CU BRAH proponemos una sencilla práctica: si conocemos de la lengua inglesa, pues son sus idiomas las implicadas por este sistema.

El MICRO COMMAND sigue dos métodos intercambiables para escuchar las palabras: uno rápido y otro lento. Si utilizamos el primer método deberemos indicarle al Sistema el tiempo que debe esperar antes de prepararse para recibir una nueva palabra, pues de no hacerlo, consideraría como una palabra completa la última mitad de la anterior.

En resumen, si todavía alguno de vosotros alberga dudas sobre la posibilidad de mantener un diálogo con nuestro micro, no tiene más que dirigirle programas o pañuelos como los descritos anteriormente, y cosa con toda seguridad, no quedará defraudado.



*EL MICRO COMMAND es un interfaz reconocedor de voz por medio de un microfono, y una cinta con el software necesario para su uso.*



# HUNTER



A gran nave interestelar S.H.1 crasó en estos momentos el sistema solar de SHADOM, perteneciente a la galaxia 37, con dirección a la Tierra.

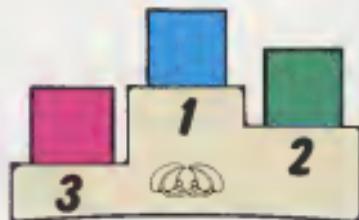
Los tripulantes de la SPACE HUNTER 1 se encuentran en situación crítica debido a una rotura en el tanque principal de combustible. Los gases emanados por el combustible, al ponerse en contacto con el aire de la nave, han provocado la explosión del sistema de turbopropulsión y como consecuencia el sobrecalentamiento de los bidones de cerveza, haciendo esta abominablemente explosiva.

A causa de estos problemas, el comandante de la nave se ve obligado a conectar la computadora de rastreo para la localización de planetas habitados, con el fin de reponer combustible y cerveza para poder continuar el viaje de regreso a la Tierra.

Como ya podremos suponer esto, aparentemente, sencilla misión no lo es tanto, y lo va a dar más de un dolor de cabeza a nuestro capitán, al margen de los peligros de la resaca de la cerveza...

## EL PROGRAMA

El programa está constituido por serie infinita de pantallas. Primeramente, el Spectrum situará de



*El objetivo del juego consiste en soltar todos los planetas en el siguiente orden: azul, verde y morado.*

forma aleatoria en la pantalla cinco planetas, simbolizados por cuadrados de tres colores diferentes:

La nave está representada por un pixel que irá dejando marcado en la pantalla su trayectoria, un incertidumbre rastro de energía que deberemos evitar.



## PROGRAMA



El objetivo del juego consiste en unir todos y cada uno de los planetas en el siguiente orden: azules, verdes y morados. En el caso en que este orden fuese roto, nuestro Spectrum respondería con un zumbido, indicando que hemos deshecho la secuencia, y por tanto nos veímos en la obligación de volver a empezar.

Por cada pantalla superada, el ordenador añadirá tres planetas más para la siguiente, aumentando así la dificultad del juego. Es muy importante recordar que si en nuestro trayecto se nos ocurriese cruzar la estela de energía que vamos trazando, pensariamos instantáneamente:



Por cada pantalla superada, el astromotor adquirirá tres planetas más, aumentando así el nivel de dificultad.



Para el control de los mandos del SPACE HUNTER 1, utilizaremos las siguientes teclas:

- O - Arriba
- A - Abajo
- O - Izquierda
- P - Derecha

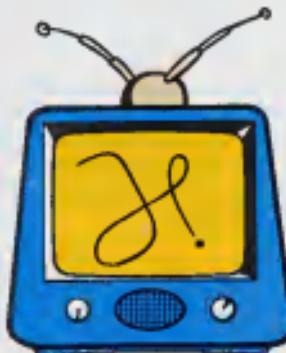
El programa únicamente incluye un gráfico definido (de la tools A), que aparece subrayado según la notación habitual en la línea 260. Así



Para conseguir pasar a otra pantalla es necesario unir todos los planetas que aparecen en los siguientes niveles:

- 1 - Planetas azules
- 2 - Planetas verdes
- 3 - Planetas morados

La SPACE HUNTER 1 está representada por un planeta que va dejando uniendo su trayectoria en la pantalla.



para, una vez introducido el programa, sólo hemos de ejecutar SAVE "S.HUNTER" o SAVE "S.HUNTER" LINE 10 para la grabación del programa en modo normal o autoejecución, respectivamente.

```

22 1000 PRINT "INTRODUCIR DATOS INICIALES"
23 1010 INPUT "JUEGO, NIVEL, DIFICULTAD"
24 1020 INPUT "NOMBRE"
25 1030 INPUT "FECHA"
26 1040 INPUT "PUNTOS TOTAL", P
27 1050 LET P=0
28 1060 LET P=0
29 1070 LET P=0
30 1080 LET P=0
31 1090 LET P=0
32 1100 LET P=0
33 1110 LET P=0
34 1120 LET P=0
35 1130 LET P=0
36 1140 LET P=0
37 1150 LET P=0
38 1160 LET P=0
39 1170 LET P=0
40 1180 LET P=0
41 1190 LET P=0
42 1200 LET P=0
43 1210 LET P=0
44 1220 LET P=0
45 1230 LET P=0
46 1240 LET P=0
47 1250 LET P=0
48 1260 LET P=0
49 1270 LET P=0
50 1280 LET P=0
51 1290 LET P=0
52 1300 LET P=0
53 1310 LET P=0
54 1320 LET P=0
55 1330 LET P=0
56 1340 LET P=0
57 1350 LET P=0
58 1360 LET P=0
59 1370 LET P=0
60 1380 LET P=0
61 1390 LET P=0
62 1400 LET P=0
63 1410 LET P=0
64 1420 LET P=0
65 1430 LET P=0
66 1440 LET P=0
67 1450 LET P=0
68 1460 LET P=0
69 1470 LET P=0
70 1480 LET P=0
71 1490 LET P=0
72 1500 LET P=0
73 1510 LET P=0
74 1520 LET P=0
75 1530 LET P=0
76 1540 LET P=0
77 1550 LET P=0
78 1560 LET P=0
79 1570 LET P=0
80 1580 LET P=0
81 1590 LET P=0
82 1600 LET P=0
83 1610 LET P=0
84 1620 LET P=0
85 1630 LET P=0
86 1640 LET P=0
87 1650 LET P=0
88 1660 LET P=0
89 1670 LET P=0
90 1680 LET P=0
91 1690 LET P=0
92 1700 LET P=0
93 1710 LET P=0
94 1720 LET P=0
95 1730 LET P=0
96 1740 LET P=0
97 1750 LET P=0
98 1760 LET P=0
99 1770 LET P=0
100 1780 LET P=0
101 1790 LET P=0
102 1800 LET P=0
103 1810 LET P=0
104 1820 LET P=0
105 1830 LET P=0
106 1840 LET P=0
107 1850 LET P=0
108 1860 LET P=0
109 1870 LET P=0
110 1880 LET P=0
111 1890 LET P=0
112 1900 LET P=0
113 1910 LET P=0
114 1920 LET P=0
115 1930 LET P=0
116 1940 LET P=0
117 1950 LET P=0
118 1960 LET P=0
119 1970 LET P=0
120 1980 LET P=0
121 1990 LET P=0
122 2000 LET P=0
123 2010 LET P=0
124 2020 LET P=0
125 2030 LET P=0
126 2040 LET P=0
127 2050 LET P=0
128 2060 LET P=0
129 2070 LET P=0
130 2080 LET P=0
131 2090 LET P=0
132 2100 LET P=0
133 2110 LET P=0
134 2120 LET P=0
135 2130 LET P=0
136 2140 LET P=0
137 2150 LET P=0
138 2160 LET P=0
139 2170 LET P=0
140 2180 LET P=0
141 2190 LET P=0
142 2200 LET P=0
143 2210 LET P=0
144 2220 LET P=0
145 2230 LET P=0
146 2240 LET P=0
147 2250 LET P=0
148 2260 LET P=0
149 2270 LET P=0
150 2280 LET P=0
151 2290 LET P=0
152 2300 LET P=0
153 2310 LET P=0
154 2320 LET P=0
155 2330 LET P=0
156 2340 LET P=0
157 2350 LET P=0
158 2360 LET P=0
159 2370 LET P=0
160 2380 LET P=0
161 2390 LET P=0
162 2400 LET P=0
163 2410 LET P=0
164 2420 LET P=0
165 2430 LET P=0
166 2440 LET P=0
167 2450 LET P=0
168 2460 LET P=0
169 2470 LET P=0
170 2480 LET P=0
171 2490 LET P=0
172 2500 LET P=0
173 2510 LET P=0
174 2520 LET P=0
175 2530 LET P=0
176 2540 LET P=0
177 2550 LET P=0
178 2560 LET P=0
179 2570 LET P=0
180 2580 LET P=0
181 2590 LET P=0
182 2600 LET P=0
183 2610 LET P=0
184 2620 LET P=0
185 2630 LET P=0
186 2640 LET P=0
187 2650 LET P=0
188 2660 LET P=0
189 2670 LET P=0
190 2680 LET P=0
191 2690 LET P=0
192 2700 LET P=0
193 2710 LET P=0
194 2720 LET P=0
195 2730 LET P=0
196 2740 LET P=0
197 2750 LET P=0
198 2760 LET P=0
199 2770 LET P=0
200 2780 LET P=0
201 2790 LET P=0
202 2800 LET P=0
203 2810 LET P=0
204 2820 LET P=0
205 2830 LET P=0
206 2840 LET P=0
207 2850 LET P=0
208 2860 LET P=0
209 2870 LET P=0
210 2880 LET P=0
211 2890 LET P=0
212 2900 LET P=0
213 2910 LET P=0
214 2920 LET P=0
215 2930 LET P=0
216 2940 LET P=0
217 2950 LET P=0
218 2960 LET P=0
219 2970 LET P=0
220 2980 LET P=0
221 2990 LET P=0
222 3000 LET P=0
223 3010 LET P=0
224 3020 LET P=0
225 3030 LET P=0
226 3040 LET P=0
227 3050 LET P=0
228 3060 LET P=0
229 3070 LET P=0
230 3080 LET P=0
231 3090 LET P=0
232 3100 LET P=0
233 3110 LET P=0
234 3120 LET P=0
235 3130 LET P=0
236 3140 LET P=0
237 3150 LET P=0
238 3160 LET P=0
239 3170 LET P=0
240 3180 LET P=0
241 3190 LET P=0
242 3200 LET P=0
243 3210 LET P=0
244 3220 LET P=0
245 3230 LET P=0
246 3240 LET P=0
247 3250 LET P=0
248 3260 LET P=0
249 3270 LET P=0
250 3280 LET P=0
251 3290 LET P=0
252 3300 LET P=0
253 3310 LET P=0
254 3320 LET P=0
255 3330 LET P=0
256 3340 LET P=0
257 3350 LET P=0
258 3360 LET P=0
259 3370 LET P=0
260 3380 LET P=0
261 3390 LET P=0
262 3400 LET P=0
263 3410 LET P=0
264 3420 LET P=0
265 3430 LET P=0
266 3440 LET P=0
267 3450 LET P=0
268 3460 LET P=0
269 3470 LET P=0
270 3480 LET P=0
271 3490 LET P=0
272 3500 LET P=0
273 3510 LET P=0
274 3520 LET P=0
275 3530 LET P=0
276 3540 LET P=0
277 3550 LET P=0
278 3560 LET P=0
279 3570 LET P=0
280 3580 LET P=0
281 3590 LET P=0
282 3600 LET P=0
283 3610 LET P=0
284 3620 LET P=0
285 3630 LET P=0
286 3640 LET P=0
287 3650 LET P=0
288 3660 LET P=0
289 3670 LET P=0
290 3680 LET P=0
291 3690 LET P=0
292 3700 LET P=0
293 3710 LET P=0
294 3720 LET P=0
295 3730 LET P=0
296 3740 LET P=0
297 3750 LET P=0
298 3760 LET P=0
299 3770 LET P=0
300 3780 LET P=0
301 3790 LET P=0
302 3800 LET P=0
303 3810 LET P=0
304 3820 LET P=0
305 3830 LET P=0
306 3840 LET P=0
307 3850 LET P=0
308 3860 LET P=0
309 3870 LET P=0
310 3880 LET P=0
311 3890 LET P=0
312 3900 LET P=0
313 3910 LET P=0
314 3920 LET P=0
315 3930 LET P=0
316 3940 LET P=0
317 3950 LET P=0
318 3960 LET P=0
319 3970 LET P=0
320 3980 LET P=0
321 3990 LET P=0
322 4000 LET P=0
323 4010 LET P=0
324 4020 LET P=0
325 4030 LET P=0
326 4040 LET P=0
327 4050 LET P=0
328 4060 LET P=0
329 4070 LET P=0
330 4080 LET P=0
331 4090 LET P=0
332 4100 LET P=0
333 4110 LET P=0
334 4120 LET P=0
335 4130 LET P=0
336 4140 LET P=0
337 4150 LET P=0
338 4160 LET P=0
339 4170 LET P=0
340 4180 LET P=0
341 4190 LET P=0
342 4200 LET P=0
343 4210 LET P=0
344 4220 LET P=0
345 4230 LET P=0
346 4240 LET P=0
347 4250 LET P=0
348 4260 LET P=0
349 4270 LET P=0
350 4280 LET P=0
351 4290 LET P=0
352 4300 LET P=0
353 4310 LET P=0
354 4320 LET P=0
355 4330 LET P=0
356 4340 LET P=0
357 4350 LET P=0
358 4360 LET P=0
359 4370 LET P=0
360 4380 LET P=0
361 4390 LET P=0
362 4400 LET P=0
363 4410 LET P=0
364 4420 LET P=0
365 4430 LET P=0
366 4440 LET P=0
367 4450 LET P=0
368 4460 LET P=0
369 4470 LET P=0
370 4480 LET P=0
371 4490 LET P=0
372 4500 LET P=0
373 4510 LET P=0
374 4520 LET P=0
375 4530 LET P=0
376 4540 LET P=0
377 4550 LET P=0
378 4560 LET P=0
379 4570 LET P=0
380 4580 LET P=0
381 4590 LET P=0
382 4600 LET P=0
383 4610 LET P=0
384 4620 LET P=0
385 4630 LET P=0
386 4640 LET P=0
387 4650 LET P=0
388 4660 LET P=0
389 4670 LET P=0
390 4680 LET P=0
391 4690 LET P=0
392 4700 LET P=0
393 4710 LET P=0
394 4720 LET P=0
395 4730 LET P=0
396 4740 LET P=0
397 4750 LET P=0
398 4760 LET P=0
399 4770 LET P=0
400 4780 LET P=0
401 4790 LET P=0
402 4800 LET P=0
403 4810 LET P=0
404 4820 LET P=0
405 4830 LET P=0
406 4840 LET P=0
407 4850 LET P=0
408 4860 LET P=0
409 4870 LET P=0
410 4880 LET P=0
411 4890 LET P=0
412 4900 LET P=0
413 4910 LET P=0
414 4920 LET P=0
415 4930 LET P=0
416 4940 LET P=0
417 4950 LET P=0
418 4960 LET P=0
419 4970 LET P=0
420 4980 LET P=0
421 4990 LET P=0
422 5000 LET P=0
423 5010 LET P=0
424 5020 LET P=0
425 5030 LET P=0
426 5040 LET P=0
427 5050 LET P=0
428 5060 LET P=0
429 5070 LET P=0
430 5080 LET P=0
431 5090 LET P=0
432 5100 LET P=0
433 5110 LET P=0
434 5120 LET P=0
435 5130 LET P=0
436 5140 LET P=0
437 5150 LET P=0
438 5160 LET P=0
439 5170 LET P=0
440 5180 LET P=0
441 5190 LET P=0
442 5200 LET P=0
443 5210 LET P=0
444 5220 LET P=0
445 5230 LET P=0
446 5240 LET P=0
447 5250 LET P=0
448 5260 LET P=0
449 5270 LET P=0
450 5280 LET P=0
451 5290 LET P=0
452 5300 LET P=0
453 5310 LET P=0
454 5320 LET P=0
455 5330 LET P=0
456 5340 LET P=0
457 5350 LET P=0
458 5360 LET P=0
459 5370 LET P=0
460 5380 LET P=0
461 5390 LET P=0
462 5400 LET P=0
463 5410 LET P=0
464 5420 LET P=0
465 5430 LET P=0
466 5440 LET P=0
467 5450 LET P=0
468 5460 LET P=0
469 5470 LET P=0
470 5480 LET P=0
471 5490 LET P=0
472 5500 LET P=0
473 5510 LET P=0
474 5520 LET P=0
475 5530 LET P=0
476 5540 LET P=0
477 5550 LET P=0
478 5560 LET P=0
479 5570 LET P=0
480 5580 LET P=0
481 5590 LET P=0
482 5600 LET P=0
483 5610 LET P=0
484 5620 LET P=0
485 5630 LET P=0
486 5640 LET P=0
487 5650 LET P=0
488 5660 LET P=0
489 5670 LET P=0
490 5680 LET P=0
491 5690 LET P=0
492 5700 LET P=0
493 5710 LET P=0
494 5720 LET P=0
495 5730 LET P=0
496 5740 LET P=0
497 5750 LET P=0
498 5760 LET P=0
499 5770 LET P=0
500 5780 LET P=0
501 5790 LET P=0
502 5800 LET P=0
503 5810 LET P=0
504 5820 LET P=0
505 5830 LET P=0
506 5840 LET P=0
507 5850 LET P=0
508 5860 LET P=0
509 5870 LET P=0
510 5880 LET P=0
511 5890 LET P=0
512 5900 LET P=0
513 5910 LET P=0
514 5920 LET P=0
515 5930 LET P=0
516 5940 LET P=0
517 5950 LET P=0
518 5960 LET P=0
519 5970 LET P=0
520 5980 LET P=0
521 5990 LET P=0
522 6000 LET P=0
523 6010 LET P=0
524 6020 LET P=0
525 6030 LET P=0
526 6040 LET P=0
527 6050 LET P=0
528 6060 LET P=0
529 6070 LET P=0
530 6080 LET P=0
531 6090 LET P=0
532 6100 LET P=0
533 6110 LET P=0
534 6120 LET P=0
535 6130 LET P=0
536 6140 LET P=0
537 6150 LET P=0
538 6160 LET P=0
539 6170 LET P=0
540 6180 LET P=0
541 6190 LET P=0
542 6200 LET P=0
543 6210 LET P=0
544 6220 LET P=0
545 6230 LET P=0
546 6240 LET P=0
547 6250 LET P=0
548 6260 LET P=0
549 6270 LET P=0
550 6280 LET P=0
551 6290 LET P=0
552 6300 LET P=0
553 6310 LET P=0
554 6320 LET P=0
555 6330 LET P=0
556 6340 LET P=0
557 6350 LET P=0
558 6360 LET P=0
559 6370 LET P=0
560 6380 LET P=0
561 6390 LET P=0
562 6400 LET P=0
563 6410 LET P=0
564 6420 LET P=0
565 6430 LET P=0
566 6440 LET P=0
567 6450 LET P=0
568 6460 LET P=0
569 6470 LET P=0
570 6480 LET P=0
571 6490 LET P=0
572 6500 LET P=0
573 6510 LET P=0
574 6520 LET P=0
575 6530 LET P=0
576 6540 LET P=0
577 6550 LET P=0
578 6560 LET P=0
579 6570 LET P=0
580 6580 LET P=0
581 6590 LET P=0
582 6600 LET P=0
583 6610 LET P=0
584 6620 LET P=0
585 6630 LET P=0
586 6640 LET P=0
587 6650 LET P=0
588 6660 LET P=0
589 6670 LET P=0
590 6680 LET P=0
591 6690 LET P=0
592 6700 LET P=0
593 6710 LET P=0
594 6720 LET P=0
595 6730 LET P=0
596 6740 LET P=0
597 6750 LET P=0
598 6760 LET P=0
599 6770 LET P=0
600 6780 LET P=0
601 6790 LET P=0
602 6800 LET P=0
603 6810 LET P=0
604 6820 LET P=0
605 6830 LET P=0
606 6840 LET P=0
607 6850 LET P=0
608 6860 LET P=0
609 6870 LET P=0
610 6880 LET P=0
611 6890 LET P=0
612 6900 LET P=0
613 6910 LET P=0
614 6920 LET P=0
615 6930 LET P=0
616 6940 LET P=0
617 6950 LET P=0
618 6960 LET P=0
619 6970 LET P=0
620 6980 LET P=0
621 6990 LET P=0
622 7000 LET P=0
623 7010 LET P=0
624 7020 LET P=0
625 7030 LET P=0
626 7040 LET P=0
627 7050 LET P=0
628 7060 LET P=0
629 7070 LET P=0
630 7080 LET P=0
631 7090 LET P=0
632 7100 LET P=0
633 7110 LET P=0
634 7120 LET P=0
635 7130 LET P=0
636 7140 LET P=0
637 7150 LET P=0
638 7160 LET P=0
639 7170 LET P=0
640 7180 LET P=0
641 7190 LET P=0
642 7200 LET P=0
643 7210 LET P=0
644 7220 LET P=0
645 7230 LET P=0
646 7240 LET P=0
647 7250 LET P=0
648 7260 LET P=0
649 7270 LET P=0
650 7280 LET P=0
651 7290 LET P=0
652 7300 LET P=0
653 7310 LET P=0
654 7320 LET P=0
655 7330 LET P=0
656 7340 LET P=0
657 7350 LET P=0
658 7360 LET P=0
659 7370 LET P=0
660 7380 LET P=0
661 7390 LET P=0
662 7400 LET P=0
663 7410 LET P=0
664 7420 LET P=0
665 7430 LET P=0
666 7440 LET P=0
667 7450 LET P=0
668 7460 LET P=0
669 7470 LET P=0
670 7480 LET P=0
671 7490 LET P=0
672 7500 LET P=0
673 7510 LET P=0
674 7520 LET P=0
675 7530 LET P=0
676 7540 LET P=0
677 7550 LET P=0
678 7560 LET P=0
679 7570 LET P=0
680 7580 LET P=0
681 7590 LET P=0
682 7600 LET P=0
683 7610 LET P=0
684 7620 LET P=0
685 7630 LET P=0
686 7640 LET P=0
687 7650 LET P=0
688 7660 LET P=0
689 7670 LET P=0
690 7680 LET P=0
691 7690 LET P=0
692 7700 LET P=0
693 7710 LET P=0
694 7720 LET P=0
695 7730 LET P=0
696 7740 LET P=0
697 7750 LET P=0
698 7760 LET P=0
699 7770 LET P=0
700 7780 LET P=0
701 7790 LET P=0
702 7800 LET P=0
703 7810 LET P=0
704 7820 LET P=0
705 7830 LET P=0
706 7840 LET P=0
707 7850 LET P=0
708 7860 LET P=0
709 7870 LET P=0
710 7880 LET P=0
711 7890 LET P=0
712 7900 LET P=0
713 7910 LET P=0
714 7920 LET P=0
715 7930 LET P=0
716 7940 LET P=0
717 7950 LET P=0
718 7960 LET P=0
719 7970 LET P=0
720 7980 LET P=0
721 7990 LET P=0
722 8000 LET P=0
723 8010 LET P=0
724 8020 LET P=0
725 8030 LET P=0
726 8040 LET P=0
727 8050 LET P=0
728 8060 LET P=0
729 8070 LET P=0
730 8080 LET P=0
731 8090 LET P=0
732 8100 LET P=0
733 8110 LET P=0
734 8120 LET P=0
735 8130 LET P=0
736 8140 LET P=0
737 8150 LET P=0
738 8160 LET P=0
739 8170 LET P=0
740 8180 LET P=0
741 8190 LET P=0
742 8200 LET P=0
743 8210 LET P=0
744 8220 LET P=0
745 8230 LET P=0
746 8240 LET P=0
747 8250 LET P=0
748 8260 LET P=0
749 8270 LET P=0
750 8280 LET P=0
751 8290 LET P=0
752 8300 LET P=0
753 8310 LET P=0
754 8320 LET P=0
755 8330 LET P=0
756 8340 LET P=0
757 8350 LET P=0
758 8360 LET P=0
759 8370 LET P=0
760 8380 LET P=0
761 8390 LET P=0
762 8400 LET P=0
763 8410 LET P=0
764 8420 LET P=0
765 8430 LET P=0
766 8440 LET P=0
767 8450 LET P=0
768 8460 LET P=0
769 8470 LET P=0
770 8480 LET P=0
771 8490 LET P=0
772 8500 LET P=0
773 8510 LET P=0
774 8520 LET P=0
775 8530 LET P=0
776 8540 LET P=0
777 8550 LET P=0
778 8560 LET P=0
779 8570 LET P=0
780 8580 LET P=0
781 8590 LET P=0
782 8600 LET P=0
783 8610 LET P=0
784 8620 LET P=0
785 8630 LET P=0
786 8640 LET P=0
787 8650 LET P=0
788 8660 LET P=0
789 8670 LET P=0
790 8680 LET P=0
791 8690 LET P=0
792 8700 LET P=0
793 8710 LET P=0
794 8720 LET P=0
795 8730 LET P=0
796 8740 LET P=0
797 8750 LET P=0
798 8760 LET P=0
799 8770 LET P=0
800 8780 LET P=0
801 8790 LET P=0
802 8800 LET P=0
803 8810 LET P=0
804 8820 LET P=0
805 8830 LET P=0
806 8840 LET P=0
807 8850 LET P=0
808 8860 LET P=0
809 8870 LET P=0
810 8880 LET P=0
811 8890 LET P=0
812 8900 LET P=0
813 8910 LET P=0
814 8920 LET P=0
815 8930 LET P=0
816 8940 LET P=0
817 8950 LET P=0
818 8960 LET P=0
819 8970 LET P=0
820 8980 LET P=0
821 8990 LET P=0
822 9000 LET P=0
823 9010 LET P=0
824 9020 LET P=0
825 9030 LET P=0
826 9040 LET P=0
827 9050 LET P=0
828 9060 LET P=0
829 9070 LET P=0
830 9080 LET P=0
831 9090 LET P=0
832 9100 LET P=0
833 9110 LET P=0
834 9120 LET P=0
835 9130 LET P=0
836 9140 LET P=0
837 9150 LET P=0
838 9160 LET P=0
839 9170 LET P=0
840 9180 LET P=0
841 9190 LET P=0
842 9200 LET P=0
843 9210 LET P=0
844 9220 LET P=0
845 9230 LET P=0
846 9240 LET P=0
847 9250 LET P=0
848 9260 LET P=0
849 9270 LET P=0
850 9280 LET P=0
851 9290 LET P=0
852 9300 LET P=0
853 9310 LET P=0
854 9320 LET P=0
855 9330 LET P=0
856 9340 LET P=0
857 9350 LET P=0
858 9360 LET P=0
859 9370 LET P=0
860 9380 LET P=0
861 9390 LET P=0
862 9400 LET P=0
863 9410 LET P=0
864 9420 LET P=0
865 9430 LET P=0
866 9440 LET P=0
867 9450 LET P=0
868 9460 LET P=0
869 9470 LET P=0
870 9480 LET P=0
871 9490 LET P=0
872 9500 LET P=0
873 9510 LET P=0
874 9520 LET P=0
875 9530 LET P=0
876 9540 LET P=0
877 9550 LET P=0
878 9560 LET P=0
879 9570 LET P=0
880 9580 LET P=0
881 9590 LET P=0
882 9600 LET P=0
883 9610 LET P=0
884 9620 LET P=0
885 9630 LET P=0
886 9640 LET P=0
887 9650 LET P=0
888 9660 LET P=0
889 9670 LET P=0
890 9680 LET P=0
891 9690 LET P=0
892 9700 LET P=0
893 9710 LET P=0
894 9720 LET P=0
895 9730 LET P=0
896 9740 LET P=0
897 9750 LET P=0
898 9760 LET P=0
899 9770 LET P=0
900 9780 LET P=0
901 9790 LET P=0
902 9800 LET P=0
903 9810 LET P=0
904 9820 LET P=0
905 9830 LET P=0
906 9840 LET P=0
907 9850 LET P=0
908 9860
```